VAXcluster System Management

Workbook

EY-9788E-WB.H001

First Edition, February 1989

CONTENTS

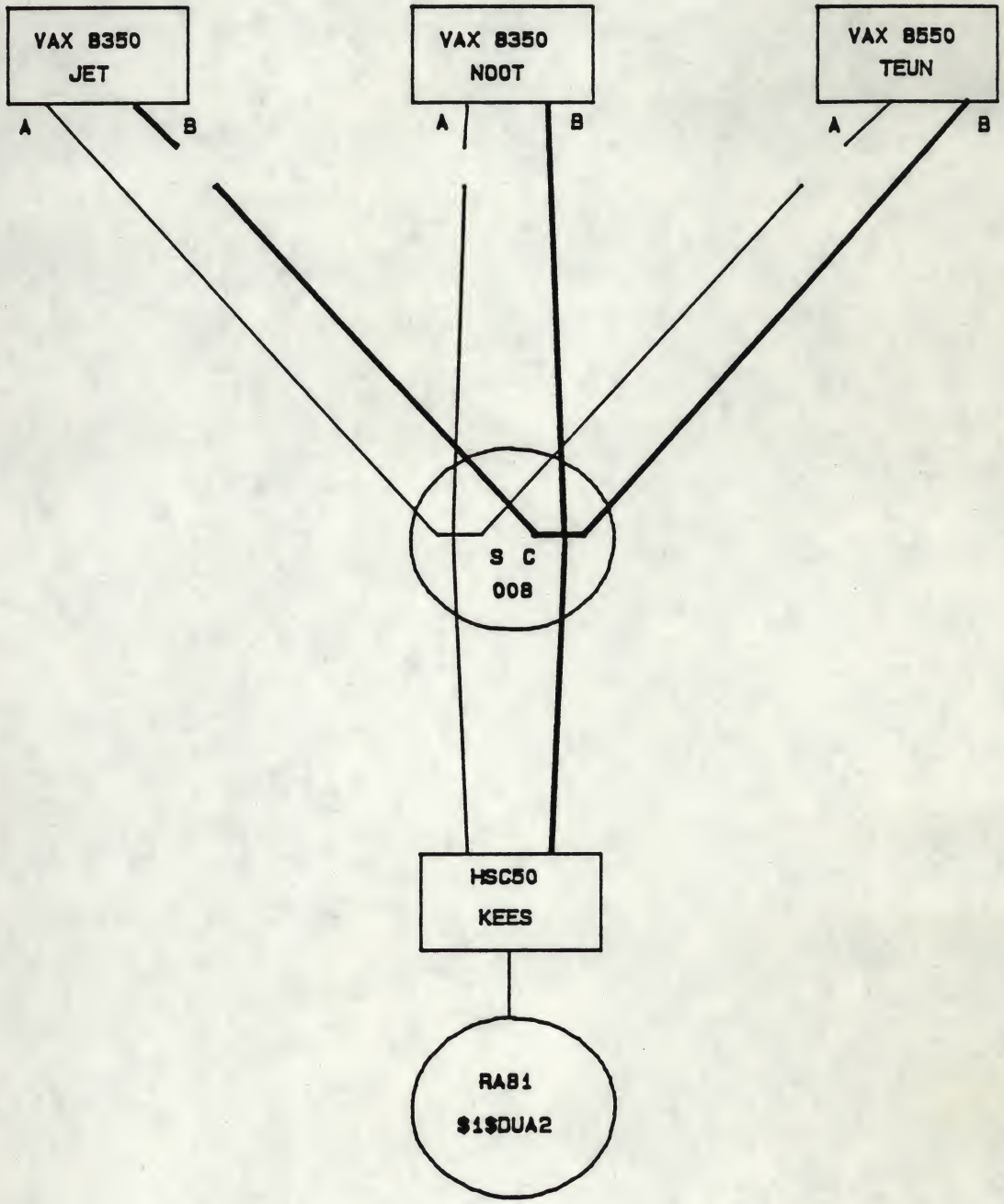
PARTITIONED VAXCLUSTER

VAXCLUSTER SYSGEN PARAMETERS

MISCELLANEOUS INFORMATION

LABORATORY EXERCISES H760

CONFIGURATION DRAWING OF A PARTITIONED VAXCLUSTER



REMOVE NODE SHUTDOWN PROCEDURE

```
$ @sys$system:shutdown

        SHUTDOWN -- Perform an Orderly System Shutdown

How many minutes until final shutdown [0]:
Reason for shutdown [Standalone]:
Do you want to spin down the disk volumes [NO]?
Do you want to invoke the site-specific shutdown procedure [YES]?
Should an automatic system reboot be performed [NO]?
When will the system be rebooted [later]:
Shutdown options (enter as a comma-separated list)
 REMOVE_NODE            Remaining nodes in the cluster should
                        adjust quorum
 CLUSTER_SHUTDOWN       Entire cluster is shutting down
 REBOOT_CHECK           Check existence of basic system files
 SAVE_FEEDBACK          Save AUTOGEN feedback information from
                        this boot

Shutdown options [NONE]: REB,REM

%SHUTDOWN-I-BOOTCHECK, Performing reboot consistency check...
%SHUTDOWN-I-CHECKOK, Basic reboot consistency check completed

%%%%%%%%%%% OPCOM    19-JAN-1989  12:37:15.14  %%%%%%%%%%%
Message from user WIEBENGA on JET
_JET$OPAO:, JET shutdown was requested by the operator.

%%%%%%%%%%% OPCOM    19-JAN-1989  12:37:16.02  %%%%%%%%%%%
Logfile was closed by operator _JET$OPA0:
Logfile was SYS$SYSROOT:[SYSMGR]OPERATOR.LOG;165

%%%%%%%%%%% OPCOM    19-JAN-1989  12:37:16.36  %%%%%%%%%%%
Operator _JET$OPA0: has been disabled, username SYSTEM

%CNXMAN,  Proposing modificati
        SYSTEM SHUTDOWN COMPLETE - USE CONSOLE TO HALT SYSTEM
```

SHUTDOWN NODE JET SEEN FROM THE VAXCLUSTER

```
%%% OPCOM 19-JAN-1989 12:37:49.10 %%%(from node JET at 19-JAN-1989 12:37:
Message from user WIEBENGA on JET
_JET$OPA0:,  JET shutdown was requested by the operator.

%%% OPCOM 19-JAN-1989 12:38:02.08 %%%(from node JET at 19-JAN-1989 12:37:
12:37:27.43 Node JET (csid 00010007) proposed modification of quorum or q
disk membership

%CNXMAN,    Removed from VAXcluster system JET
%CNXMAN,    Lost connection to system JET
%CNXMAN,    Quorum lost, blocking activity
%CNXMAN,    Timed-out lost connection to system JET
%%% OPCOM 19-JAN-1989 12:38:02.12 %%% (from node JET at
            19-JAN-1989 12:37:27.51)
12:37:27.45 Node Jet (csid 00010007) completed VAXcluster state
            transition

%%% OPCOM 19-JAN-1989 12:38:44.34 %%%
12:38:02.25 Node Jet (csid 00010007) has been removed from the vaxcluster

%%% OPCOM 19-JAN-1989 12:38:44.39 %%%
12:38:02.25 Node NOOT (csid 0001000A) lost connection to node JET

%%% OPCOM 19-JAN-1989 12:38:44.42 %%%
12:38:02.25 Node NOOT (csid 0001000A) lost quorum,blocking activity

%%% OPCOM 19-JAN-1989 12:38:44.44 %%%
12:38:02.26 Node NOOT (csid 0001000A) timed out lost connection to node J

%%% OPCOM 19-JAN-1989 12:38:44.46 %%%
12:38:44.16 Node NOOT (csid 0001000A) regained quorum,proceeding
```

BOOT NODE JET WITH ITS OWN SYSTEM DISK

```
>>>BOOT

       VAX/VMS Version V5.02  08-DEC-1988 20:00

%PAA0, PATH #1. Has gone from GOOD to BAD - REMOTE PORT    0

%PAA0, PATH #1. Has gone from GOOD to BAD - REMOTE PORT    1

waiting to form or join VAXcluster
%CNXMAN,  Proposing formation of a VAXcluster
%CNXMAN,  Now a VAXcluster member -- system JET
%CNXMAN,  Completing VAXcluster state transition

%%%%%%%%%%  OPCOM   19-JAN-1989 13:00:38.91  %%%%%%%%%%
Logfile has been initialized by operator _JET$OPA0:
Logfile is SYS$SYSROOT:[SYSMGR]OPERATOR.LOG;6

%%%%%%%%%%  OPCOM   19-JAN-1989 13:00:40.62  %%%%%%%%%%
12:59:38.26 Node JET (csid 00000000) proposed formation of a vaxcluster

%%%%%%%%%%  OPCOM   19-JAN-1989 13:00:40.69  %%%%%%%%%%
12:59:38.26 Node JET (csid 00010007) is now a vaxcluster member

%%%%%%%%%%  OPCOM   19-JAN-1989 13:00:40.76  %%%%%%%%%%
12:59:38.36 Node JET (csid 00010007) completed VAXcluster state transitio

|
|

%SET-I-INTSET, login interactive limit = 64, current interactive
value = 0
  SYSTEM       job terminated at 19-JAN-1989 13:01:35.37
```

PARTITIONED VAXCLUSTER

SHOW CLUSTER OUTPUT FROM NODE IN PARTITIONED VAXCLUSTER

SYSTEMS	MEMBERS			CIRCUIT
NODE	VOTES	EXP_VO	STATUS	CABLE
JET	1	1	MEMBER	A - B
DOES				A -
KEES				A -

CLUSTER			
CL_QUORUM	CL_VOTES	FORMED	LAST_TRANSITION
1	1	19-JAN-89 12:59	19-JAN-89 12:59

SHOW CLUSTER OUTPUT FROM PARTITIONED VAXCLUSTER

View of Cluster from system ID 41133 node: NOOT

SYSTEMS		MEMBERS	CIRCUITS				
NODE	HW_TYPE	STATUS	RPORT	RP_TYP	CIR_STA	CABLE	RP_REV
NOOT	VAX 8350	MEMBER	3	CIBCA	OPEN	A - B	20003
DOES	HS50		1	HSC50	OPEN	- B	225
TEUN	VAX 8550	MEMBER	2	CIBCI	OPEN	- B	80007
JET	VAX 8350	BRK_NO					
KEES	HS50		0	HSC50	OPEN	- B	225

CLUSTER		
QD_NAME	FORMED	LAST_TRANSITION
1DUA2	10-JAN-89 17:26	19-JAN-89 12:53

LOCAL
NAME
PAAO

BOOT NODE JET IN PARTITIONED VAXCLUSTER (WITH QUORUM DISK)

```
>>>BOOT

  VAX/VMS Version V5.02  8-DEC-1988 20:00

%PAA0, PATH #1. Has gone from GOOD to BAD - REMOTE PORT   0

%PAA0, PATH #1. Has gone from GOOD to BAD - REMOTE PORT   1

waiting to form or join VAXcluster
%CNXMAN,  Estabished "connection" to quorum disk

                !       Comment  The "VAXcluster" JET succeeds
                !       to read the Quorum.dat on the
                !       quorum disk. But is not allowed to
                !       add the QDvotes for quorum calculation,
                !       by means of the excisting
                !       VAXcluster in which the quorum
                !       disk allready a member is.
                !       Node JET just waits and sits at this
                !       point. The only thing to do here
                !       is control/P node JET.

80008B1F  02
>>>
```

NODE JET JOINS THE VAXCLUSTER

```
>>>BOOT

    VAX/VMS VERSION V5.02 8-DEC-1988 20:00


%CNXMAN,  Discovered system NOOT
%CNXMAN,  Established connection to system NOOT
%CNXMAN,  Discovered system TEUN
%CNXMAN,  Established connection to system TEUN
waiting to form or join VAXcluster
%CNXMAN,  Sending VAXcluster membership request to system TEUN
%CNXMAN,  Now a VAXcluster member -- system JET
%CNXMAN,  Established "connection" to quorum disk
%CNXMAN,  Proposing modification of quorum or quorum disk
membership
%CNXMAN,  Completing VAXcluster state transition
%%%%%%%%%%  OPCOM  19-JAN-1989 14:15.45  %%%%%%%%%%
Logfile has been initialized by operator _JET$OPA0:
Logfile is SYS$SYSROOT:[SYSMGR]OPERATOR.LOG;166

%%%%%%%%%%  OPCOM  19-JAN-1989 14:15.52.66  %%%%%%%%%%
14:14:17.03 Node JET (csid 0001000E) is now a VAXcluster member

%%%%%%%%%%  OPCOM  19-JAN-1989 14:15.52.75  %%%%%%%%%%
14:14:17.81 Node JET (csid 0001000E) re-established connection to
quorum disk

%%%%%%%%%%  OPCOM  19-JAN-1989 14:15.52.84  %%%%%%%%%%
14:14:17.81 Node JET (csid 0001000E) proposed modification of
quorum or quorum disk membership

%%%%%%%%%%  OPCOM  19-JAN-1989 14:15.52.90  %%%%%%%%%%
14:14:17.83 Node JET (csid 0001000E) completed VAXcluster state
transition
```

NODE JET JOINS THE VAXCLUSTER SEEN FROM EXCISTING VAXCLUSTER

```
$
$
%CNXMAN,  Deleting CSB for system JET
%CNXMAN,  Discovered system JET
%CNXMAN,  Established connection to system JET
%%% OPCOM 19-JAN-89 14:14:42.86 %%%
14:14:42.85 Node NOOT (sysid 41133) discovered node JET (sysid
41134)

%%% OPCOM 19-JAN-89 14:14:42.94 %%%
14:14:42.85 Node NOOT (csid 0001000A) established connection to
node JET

%%% OPCOM 19-JAN-89 14:14:43.63 %%%(from node TEUN at 19-JAN-89
14:14:43.15)
14:14:43.54 node TEUN (sysid 51371) discovered node JET
(sysid 41134)

%%% OPCOM 19-JAN-89 14:14:43.69 %%%(from node TEUN at 19-JAN-89
14:14:43.61)
14:14:43.54 node TEUN (csid 0001000D) established connection to
node JET

%%% OPCOM 19-JAN-89 14:14:52.38 %%%(from node TEUN at 19-JAN-89
14:14:52.29)
14:14:52.29 node TEUN (csid 0001000D) received VAXcluster
membership request from node JET

%%% OPCOM 19-JAN-89 14:14:52.43 %%%(from node TEUN at 19-JAN-89
14:14:52.34)
14:14:52.29 node TEUN (csid 0001000D) proposed addition of node
JET

%%% OPCOM 19-JAN-89 14:14:52.65 %%%(from node TEUN at 19-JAN-89
14:14:52.48)
14:14:52.47 node TEUN (csid 0001000D) completed VAXcluster state
transition
```

SHOW CLUSTER OUTPUT, NODE JET JOINING THE VAXCLUSTER

View of Cluster from system ID 41133 node: NOOT

SYSTEMS		MEMBERS	CIRCUITS				
NODE	HW_TYPE	STATUS	RPORT	RP_TYP	CIR_STA	CABLE	RP_REV
NOOT	VAX 8350	MEMBER	3	CIBCA	OPEN	A - B	20003
DOES	HS50		1	HSC50	OPEN	A - B	225
TEUN	VAX 8550	MEMBER	2	CIBCI	OPEN	A - B	80007
JET	VAX 8350	NEW	4	CIBCA	OPEN	A - B	20003
KEES	HS50		0	HSC50	OPEN	A - B	225

CLUSTER			LOCAL
QD_NAME	FORMED	LAST_TRANSITION	NAME
1DUA2	10-JAN-89 17:26	19-JAN-89 13:12	PAAO

VAXCLUSTER SYSGEN PARAMETERS

INTRODUCTION

This chapter contains a selection of SYSGEN parameters.

PARAMETER DESCRIPTIONS

ACP_REBLDSYSD

ACP_REBLDSYSD specifies whether the system disk should be rebuilt if it was improperly dismounted with extent caching, file number caching, or disk quota caching enabled. The ACP_REBLDSYSD default value (1) ensures that the system disk is rebuilt.

Depending on the amount of caching enabled on the volume before it was dismounted, the rebuild operation may consume a considerable amount of time. Setting the value of ACP_REBLDSYSD to 0 specifies that the disk should be returned to active service immediately. If you set ACP_REBLDSYSD to 0, you can enter the DCL command SET VOLUME/REBUILD at any time to rebuild the disk.

ALLOCLASS

ALLOCLASS determines the device allocation class for the system. The device allocation class is used to derive a common lock resource name for multiple access paths to the same device.

DUMPSTYLE

DUMPSYLE specifies the method of writing system dumps. Specify one of the following values:

Value	Meaning
0	The entire contents of physical memory will be written to the dump file. This is the default.
1	Selective portions of memory will be written to the dump file as space permits.

If you have a large memory system and the dump file is too small to contain a complete system dump of physical memory, set DUMPSTYLE to 1 to specify a partial memory dump.

EXPECTED_VOTES

EXPECTED_VOTES specifies the maximum number of votes that may be present in a VAXcluster at any given time. Set it to a value that is equal to the sum of the vote parameters of all VAXcluster members, plus any votes that are contributed by the quorum disk. This value is used to automatically derive the number of votes that must be present for the VAXcluster to function (quorum).

LOCKDIRWT

LOCKDIRWT determines the portion of lock manager directory that will be handled by this system. The default value is usually adequate.

LOCKIDTBL (M)

LOCKIDTBL sets initial number of entries in the system Lock ID table and defines the amount by which the Lock ID table is extended whenever the system runs out of locks. There must be one entry for each lock in the system; each entry requires four bytes.

MSCP_BUFFER

MSCP_BUFFER specifies the number of pages to be allocated to the MSCP server's local buffer area. This buffer area is the space used be the server to transfer data between client systems and local disks.

MSCP_CREDITS

MSCP_CREDITS specifies the number of outstanding I/O requests that can be active from one client system.

MSCP_LOAD

MSCP_LOAD controls the loading of the MSCP server during a system boot. Specify one of the following values:

Value	Meaning
0	Do not load the MSCP server. This is the default value.
1	Load the MSCP server and serve disks as specified by the MSCP_SERVE_ALL parameter.

MSCP_SERVE_ALL

MSCP_SERVE_ALL controls the serving of disks during a system boot. Specify one of the following values:

Value	Meaning
0	Do not serve any disks. This is the default.
1	Serve all available disks.
2	Serve only locally-attached (non-HSC) disks.

If the MSCP_LOAD system parameter is zero, MSCP_SERVE_ALL is ignored.

MVTIMEOUT (D)

MVTIMEOUT is the time in seconds that a mount verification attempt continues on a given disk vlume. If the mount verification does not recover the volume within that time, the I/O operations outstanding to the volume terminate abnormally.

NISCS_CONV_BOOT

NISCS_CONV_BOOT controls whether or not a conversational boot is permitted during a remote system boot. The default of 0 specifies that conversational boots are not permitted.

NISCS_LOAD_PEAO

NISCS_LOAD_PEAO controls whether or not the NI-SCS port driver PEDRIVER is loaded during system boot. The default of 0 specifies that the PEDRIVER is not loaded.

NISCS_PORT_SERV

NISCS_PORT_SERV provides flag bits for PEDRIVER port services. Bits 0 and 1 set (decimal value 3) enables data checking. The remaining bits are reserved for future use.

PAMAXPORT (D)

PAMAXPORT specifies the maximum port number that the CI port driver polls to discover newly initialized ports of failed remote ports.

You can decrease this parameter to reduce polling activity if the hardware configuration has fewer than 16 ports. For example, if the configuration has a total of 5 ports assigned to port numbers 0 through 4, you could set PAMAXPORT to 4.

If no CI device is configured on your system, this parameter is ignored.

PANOPOLL (D)

PANOPOLL suppresses CI polling for ports. If PANOPOLL is set to 1, a VAXcluster member node does not discover that another member node has shut down or powered down quickly or that a new member node has booted. This parameter is useful if you want to bring up a system that is isolated from the rest of the VAXcluster for checkout purposes. Setting PANOPOLL is equivalent to uncabling the system from the star coupler. The default value of 0 (off) is the normal setting and is required if you are booting from an HSC or if your system is joining a VAXcluster.

If no CI device is configured on your system, this parameter is ignored.

PANUMPOLL (D)

Establishes the number of ports to poll each polling interval. The normal setting for PANUMPOLL is 16. The parameter is useful in applications sensitive to the amount of contiguous time that VMS spends at IPL 8 during each polling interval, while increasing the number of polls needed to discover new or failed ports.

If no CI device is configured on your system, this parameter is ignored.

PAPOLLINTERVAL (D)

PAPOLLINTERVAL specifies in seconds the polling interval the CI port driver uses to poll for a newly booted system, a broken port-to-port virtual circuit, or a failed remote port.

This parameter trades faster response to virtual circuit failures against increased polling overhead. DIGITAL recommends that you use the default value for this parameter.

If no CI device is configured on your system, this parameter is ignored.

PAPOOLINTERVAL (D)

PAPOOLINTERVAL is the interval in seconds after which a CI or UDA port driver's suspended request for message buffer allocation from nonpaged pool is awakened to repeat the request. A request is suspended if there is unsufficient nonpaged pool.

If no CI device or UDA 50/52 is configured on your system, this parameter is ignored.

The default value should always be adequate.

PASANITY (D)

PASANITY controls whether the port sanity timer is enabled to permit remote systems to detect a system that has been hung at IPL 8 or above for 100 seconds. This parameter is normally set to 1 and should be set to 0 only when you are debugging with XDELTA or planning to halt the CPU for periods of 100 seconds or more.

PASANITY is only semi-dynamic. A new value of PASANITY takes effect on the next CI port reinitialization.

If no CI device is configured on your system, this parameter is ignored.

PASTDGBUF

PASTDGBUF is the number of datagram receive buffers to queue initially for the CI port driver's configuration poller; the initial value is expanded during system operation, if needed.

If no CI device is configured on your system, this parameter is ignored.

PASTIMOUT (D)

PASTIMOUT is the basic interval at which the CI port driver wakes up to perform time-based bookkeeping operations. It is also the period after which a start handshake datagram is assumed to have timed out.

If no CI device is configured on your system, this parameter is ignored.

The default value should always be adequate.

QDSKINTERVAL

QDSKINTERVAL establishes, in seconds, the disk quorum polling interval.

QDSKVOTES

QDSKVOTES specifies the number of votes contributed by a quorum disk in a VAXcluster.

QUORUM

This parameter is obsolete with VMS Version 5.0. VMS automatically calculates cluster quorum from the value of the EXPECTED_VOTES parameter. See the description of the EXPECTED_VOTES parameter for more information.

RECNXINTERVAL (D)

RECNXINTERVAL establishes the polling interval, in seconds, during which to attempt reconnection to a remote system.

SCSBUFFCNT (G)

SCSBUFFCNT is the number of CI buffer descriptors configured for all CI ports on the system. If no CI device is configured on your system, this parameter is ignored.

SCSCONNCNT (G)

SCSCONNCNT is the initial number of SCS connections that are configured for use by all system applications, including the one used by Directory Service Listen. The initial number will be expanded by the system if needed.

If no CI device or UDA 50/52 is configured on your system, this parameter is ignored.

The default value is adequate for all CI/UDA hardware combinations available with VMS Version 5.0.

SCSFLOWCUSH (D)

SCSFLOWCUSH is an SCS flow control parameter for sequenced messages. For each connection, SCS tracks the number of receive buffers available and communicates the number to the SCS at the remote end of the connection. However, SCS does not need to do this for each new receive buffer. Instead, SCS notifies the remote SCS of new receive buffers if the number already communicated to the remote SCS falls as low as the value of SCSFLOWCUSH.

If no CI device is configured on your system, this parameter is ignored.

SCSMAXDG (G)

SCSMAXDG is the maximum number of bytes of application data in one datagram. The amount of physical memory consumed by one datagram packet is SCSMAXDG plus overhead for buffer management.

DECnet is a primary user of SCS datagrams. For performance reasons, SCSMAXDG should be set to the same value (up to a maximum of 985) as the DECnet NCP parameter BUFFER SIZE in the executor database. (Note the maximum value for NCP parameter BUFFER SIZE is greater than the maximum value for SCSMAXDG).

If no CI device is configured on your system, tnis parameter is ignored.

SCSMAXMSG (G)

SCSMAXMSG is the maximum number of bytes of application data in one message.

If no CI device is configured on your system, this parameter is ignored.

Do not change the default value.

SCSNODE (G)

SCSNODE is the SCS system name. It should be the same as the DECnet node name (limited to six characters), since the name must be unique among all systems in the VAXcluster. Specify the parameter value as an ASCII string enclosed in parentheses. Note that the string may not include dollar sign ($) or underscore (_) characters.

SCSRESPCNT (G)

SCSRESPCNT is the total number of response descriptor table entries (RDTEs) configured for use by all system applications.

If no CI device or UDA 50/52 is configured on your system, this parameter is ignored.

SCSSYSTEMID (G)

SCSSYSTEMID specifies the lower-order 32 bits of the 48-bit system identification number. It is the unique identifier of each system and is calculated as follows:

(DECnet area number * 1024) + DECnet-VAX node number.

For example, if the DECnet address is 2.211, then SCSSYSTEMID should be set to (2 * 1024) + 211.

SCSSYTEMIDH (G)

SCSSYSTEMIDH specifies the high-order 16 bits of the 48-bit system identification number and must be set to 0.

SETTIME

SETTIME enables or disables solicitation of the time of day each time the system is booted. This parameter should usually be off (0), so that the system sets the time of day at boot time to the value of the processor time-of-day register. You can reset the time after the system is up with the DCL command SET TIME (see the VMS DCL Dictionary).

SHADOWING

SHADOWING is a Boolean value specifying the type of disk class driver that is loaded on the system. The default value of 0 loads the normal disk class driver, DUDRIVER. A value of 1 loads the shadowing disk class driver, DSDRIVER.

VAXCLUSTER

VAXCLUSTER controls loading of the cluster code. Specify one of the following:

Value	Meaning
0	Never load
1	Load if SCSLOA is being loaded
2	Always load (and also load SCSLOA)

The default value is 1.

VOTES

VOTES establishes the number of votes a VAXcluster member system contributes to a quorum.

SHOW CLUSTER UTILITY DEFAULT KEYPAD

GOLD	HELP	REFRESH	INIT
SET FUNC PAN	SET FUNC SCROLL	SET FUNC MOVE	SET FUNC EDIT
ADD	REMOVE	SET AUTO POS ON/OFF	
SET	SAVE	WRITE	
		SELECT	

BOOT A NODE

1 Classroom:

a. Find out which commands you have to give on the console in order to boot the appointed node. (Note: R1=8 ,BI node number of the CIBCA in JET)

b. Analyze whether the leaving of a node or the quorumdisk will influence the VAXcluster. (Think about the CL_QUORUM and EXPECTED_VOTES)

CONFIGURE A CI-CLUSTERNODE AND BOOT FROM IT

Classroom:

2 a. Execute the following commandprocedure in order to create your own systemroot:
- $ @SYS$COMMON:[SYSMGR]CLUSTER_CONFIG.COM

Computerroom:

b. Shutdown the VAX and boot it conversational from the just created root and examine the cluster parameters
(SYSBOOT> SHOW/CLUSTER etc.
Have a look at the parameter VAXCLUSTER. Can you find a reason for this setting ?)

c. Boot the system through by giving
SYSBOOT> CONTINU

d. Login and check if the logical SYS$SYSROOT shows your root

e. Leave the system for the next group.

BOOT AN HSC

3 a. Analyze in how many ways you can achieve a disk failover and in how many ways you can force the HSC to boot

b. On the console of DOES ,type CTRL_Y and
HSC50 > RUN SETSHO
SETSHO> SHOW ALL

c. What is the status of the connected disks ?

d. Boot the HSC DOES in one of the possibilities found in question a.

Note: Prepare question a. in the classroom and try to boot the HSC DOES as quick as possible, because another is booted over DOES as well.

MONITOR AN HSC

4 a. Let VTDPY run on HSC KEES. (HSC50> RUN DD1:VTDPY)
Which systems can you find as clustermembers,
according to KEES ?

b. Find out which utilities and files there are on the
TU58 (System and Utilities Tape)

c. If only the disk 1DUA4 gave some flaky data problems,
which port on which requestor would you point out
as the one that could cause these problems ?

d. Can you give an alternative requestor and port to
connect, the diskcable from 1DUA4, to ?

Note: Do not boot this HSC KEES at anytime, because at this
moment the complete cluster is depending on this HSC.

NI-CLUSTER, ADDING AND BOOTING A SATELLITE

5 a. Find out what the ethernet_hardware address and the node_number is of the satellite.
How many ways are there to find the ethernet hardware address ?

b. Login on the bootnode and execute the following in order to define the boot node and create a systemroot:

```
$ @SYS$MANAGER:CLUSTER_CONFIG    !  Answer to the
                                 !  questions.
```

c. Check the inputs you have to give on the console terminal from the satellite, in order to boot from this created satelliteroot.

6 a. Boot the satellite conversational .
While the satellite boots, have a look on the bootnode to SHOW CLUSTER output.

b. Examine some parameters like

```
SYSBOOT> SHOW/CLUSTER
       > SHOW/SCS
       > SHOW/SPECIAL
```

c. Boot the system through by giving the command

```
SYSBOOT> CONTINU
```

7 Shutdown the satellite.

8 Remove the satellite's root permanently.

CREATE A COMMANDPROCEDURE GETTING CLUSTERINFO

9 Create a command procedure in order to get some cluster info without using the SHOW CLUSTER utility.
You can make use of the lexical functions F$GETSYI() and F$GETDVI().

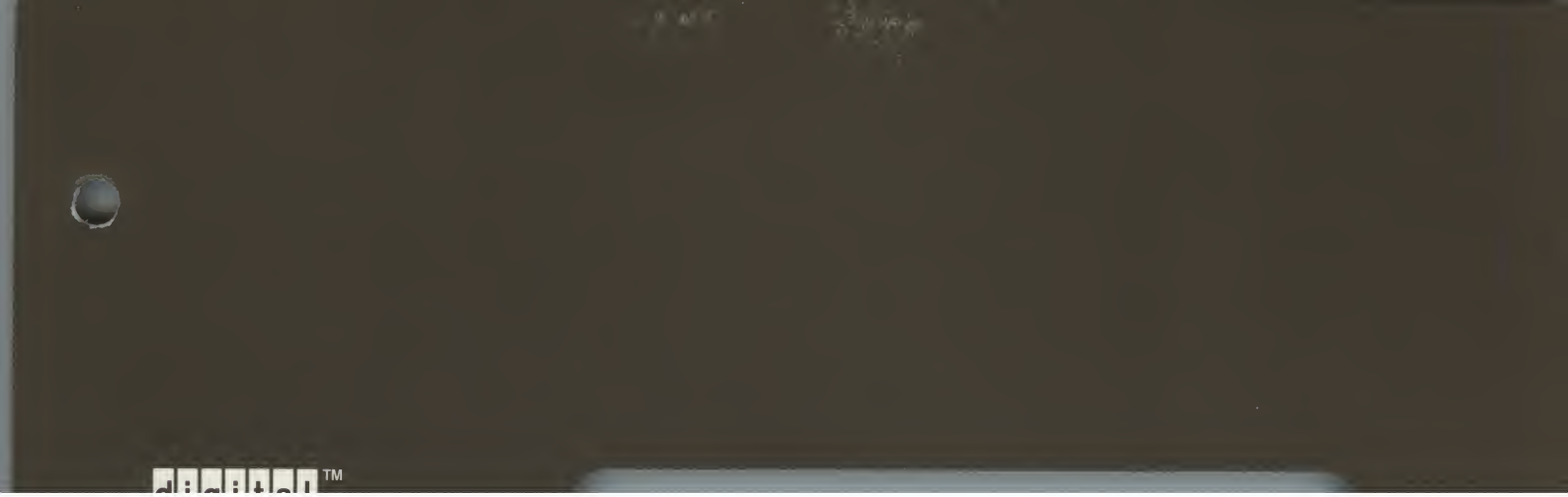
Example:

```
$ @clusterinfo.com

NODE     TYPE      EXP_VOT    VOTES

JET      8350      0003       0001
NOOT     8350      0003       0001
TEUN     8550      0003       0001

CLUSTER_QUORUM     : 3
CLUSTER_VOTES      : 5
$
```

Notes: Also be aware of the fact that a node might be out of the cluster. Detect that as well and give a message if so.

digital™

Printed in The Netherlands


digital™

VAXcluster System Management

Laboratory exercises

EY-9788E-EX.0004

Educational Services

digital™

VAXcluster System Management Laboratory Exercises

EY-9788E-EX-0004

This document was prepared and published by Educational Services Development and Publishing, Digital Equipment Corporation.

CONTENTS

LABORATORY EXERCISES

The laboratory exercises are meant to provide hands-on practice in managing a VAXcluster system. Some exercises should only be performed on a standalone cluster. These include exercises that:

- Require that you build parts of a cluster
- Are disruptive to processing on a running cluster

Some exercises do not require a standalone machine and can be performed on any cluster. Your instructor will choose which labs you can perform on your lab cluster, and may modify certain exercises to make them more suitable for your lab situation.

BUILDING A VAXcluster SYSTEM

Laboratory Exercise 1

For this exercise, your instructor must provide a scratch disk. You will build a new VMS system disk on this scratch disk and create system roots to allow several processors to boot from the disk.

Your instructor will probably divide the class into groups. Each group should perform the following steps:

1. Your instructor will tell you what nodes are in your cluster. Fill in the following tables:

Node Name	CI Port Number	DECnet Address	Hardware Type	Satellite?	System Disk Device

Node Name	System Root	Boot Server	Disk Server	HSC Disk Server	Disk Allocation Class

Node Name	System Root	Boot Server	Disk Server	HSC Disk Server	Disk Allocation Class

Node Name	Quorum Disk?	Page and Swap Device	Conversational Bootstrap?	Ethernet Hardware Address

2. Use SYS$MANAGER:CLUSTER_CONFIG.COM CREATE to make the scratch disk into a system disk.

3. Each student in the group should run CLUSTER_CONFIG.COM to add a system root to the system disk (in other words, add a new node to the cluster). Choose unique node names and DECnet addresses; your instructor may tell you to use particular DECnet addresses if the cluster is part of a larger network.

4. Create startup command procedures SYSTARTUP_V5.COM and SYLOGICALS.COM. For each procedure, your group may decide to create a single procedure that contains conditional statements and runs on every node of the cluster, or a separate procedure for each node, or a combination of the two. You may also decide to create additional procedures that are called from SYSTARTUP_V5.COM.

   Your procedures must do the following tasks:

   - Mount all HSC disks on every node of the cluster.
   - For each node that has a local disk, mount a local disk cluster-wide.
   - Create additional logical names for some of the disks that are mounted cluster-wide.
   - Start an execution queue for a printer (or a terminal set up for printing) on each node.
   - Start an execution batch queue on each node.
   - Start a generic print queue that feeds all the execution print queues.
   - Start a generic batch queue that feeds all the execution batch queues.
   - Create additional logical names for some of the queues.
   - Set the characteristics of any local terminals.

5. If your node uses a bootstrap command procedure, create a procedure that will boot your node from the root you created. Also create a procedure that performs a conversational boot from the same root.

6. If your instructor can provide you with standalone time on the cluster, boot the cluster nodes from the new system disk you created.

   - If your node uses a bootstrap command procedure, you must first copy the procedure you created onto the node's console volume.
   - Verify that your startup command procedures executed correctly.

Solutions

See your instructor if you need help with this exercise.

Laboratory Exercise 2

This exercise deals with a common problem encountered when adding one or more systems to an existing cluster. Most clusters are managed using a single UAF file which is available on a cluster-wide basis. There is a high probability that combining UAF files will reveal that there are multiple users with the same user name or UIC.

1. Copy the files V54_CLUMGT:SAMPLEA_UAF.DAT and V54_CLUMGT:SAMPLEB_UAF.DAT to your own directory.

2. Merge the two files into a single UAF that contains no duplicate user names or UICs. Follow the steps in Module 5, Building a VAXcluster System.

3. Optional: Write a command procedure that helps to merge two UAF files into a single UAF with no duplicate user names. This procedure should do the following tasks:

   a. Use the CONVERT utility to merge the files.

   b. Use the CONVERT utility to convert the exception file from sequential to indexed, so it can be examined using the AUTHORIZE utility.

   c. Use the AUTHORIZE utility to produce a listing of the converted exception file.

   d. Instruct the user to resolve changes (by using AUTHORIZE to modify the exception file, then merging the exception file with the previously merged UAF).

Solutions

1. `COPY V54_CLUMGT:SAMPLE%_UAF.DAT *`

2. See your instructor if you need the solution to this exercise.

3. Here is a command procedure that does the necessary tasks.

```
$!
$! MERGE_UAF.COM
$!
$! This procedure provides a template to merge two UAF files
$!
$ COMMON_UAF = F$DIRECTORY() + "COMMON_UAF.DAT"
$ EXCEPTION_FILE = F$DIRECTORY() + "DUP_UAF.SEQ"
$ EXCEPTION_IDX = F$DIRECTORY() + "DUP_UAF.IDX"
$ UAF_FDL = F$DIRECTORY() + "UAF.FDL"
$PROMPT1:
$ READ/PROMPT="Location of first UAF FILE:  " SYS$COMMAND UAF1
$ IF (UAF1 .EQS. "") THEN GOTO PROMPT1
$ OPEN /READ /ERROR=PROMPT1 FILE1 'UAF1'
$ CLOSE FILE1
$PROMPT2:
$ READ/PROMPT="Location of second UAF FILE:  " SYS$COMMAND UAF2
$ IF (UAF2 .EQS. "") THEN GOTO PROMPT2
$ OPEN /READ /ERROR=PROMPT2 FILE2 'UAF2'
$ CLOSE FILE2
$!
$! Generate a listing file for each UAF file
$!
$ DEFINE /USER SYSUAF 'UAF1'
$ RUN SYS$SYSTEM:AUTHORIZE
LIST
$ PRINT /DELETE SYSUAF.LIS;
$ DEFINE /USER SYSUAF 'UAF2'
$ RUN SYS$SYSTEM:AUTHORIZE
LIST
$ PRINT /DELETE SYSUAF.LIS;
$!
$! Merge the two UAF files and generate an exception file
$!
$ CONVERT 'UAF1','UAF2' 'COMMON_UAF'/EXCEPTION='EXCEPTION_FILE'
$!
$! Generate an FDL file so that we can convert the sequential
$!      exception file to an indexed file that AUTHORIZE can read.
$!
$ ANALYZE/RMS_FILE/FDL 'UAF1'/OUTPUT='UAF_FDL'
$!
$! Now convert the exception file to INDEXED
$!
$ CONVERT/FDL='UAF_FDL' 'EXCEPTION_FILE' 'EXCEPTION_IDX'
$!
$! Create a listing of the exception file
$!
$ DEFINE/USER SYSUAF 'EXCEPTION_IDX'
$ RUN SYS$SYSTEM:AUTHORIZE
LIST
$ PRINT/DELETE SYSUAF.LIS;
$ WRITE SYS$OUTPUT " "
$ WRITE SYS$OUTPUT "At this point the following files exist:"
$ WRITE SYS$OUTPUT " "
$ WRITE SYS$OUTPUT "UAF file 1:              ",UAF1
```

```
$ WRITE SYS$OUTPUT "UAF file 2:              ",UAF2
$ WRITE SYS$OUTPUT "The merged UAF file"
$ WRITE SYS$OUTPUT "(Without duplicate
$ WRITE SYS$OUTPUT "user names resoved):     ",common_UAF
$ WRITE SYS$OUTPUT "The UAF exception file: ",exception_idx
$ TYPE SYS$INPUT
```

Using the listing files generated for the merged UAF file and the exception UAF records file, make changes to the exception index file by using the following commands:

```
$ WRITE SYS$OUTPUT "$DEFINE SYSUAF ",EXCEPTION_IDX
$ WRITE SYS$OUTPUT "$RUN SYS$SYSTEM:AUTHORIZE"
$ TYPE SYS$INPUT
```

Use the COPY command to change user names in the exception index file. When all the duplicate records have been resolved, type the following command to merge the exception file with the common UAF file:

```
$ WRITE SYS$OUTPUT "$CONVERT ",COMMON_UAF,",",EXCEPTION_IDX -
 ," /OUTPUT=SYSUAF.DAI"
$ TYPE SYS$INPUT
```

If this was done correctly and all duplicate records were removed, no DUPLICATE key messages will be issued from CONVERT. If not, resolve the remaining duplicates and convert the file again.

```
$exit
ET:
$WRITE SYS$OUTPUT "FILE NOT FOUND"
```

MANAGING A VAXcluster SYSTEM

Laboratory Exercise 1

The SYSMAN utility allows the system manager to perform system management operations on all nodes of a cluster or a selected subset of nodes.

1. Use the SYSMAN utility to:

   a. Display SHOW SYSTEM output from all nodes

   b. Display SHOW USERS output from all nodes

   c. Display SHOW CLUSTER output from all nodes

   d. Display the values of the cluster-related SYSGEN parameters on all nodes

   e. With the NCP utility, list the executor characteristics of each node

   f. Display the disk quota information for all users on your default device

2. Create a small DCL command procedure that uses SYSMAN to show the time on all nodes.

3. Create a command procedure SHOW_DISK.COM with the following statements:

```
$ INQUIRE DISK_NAME "Show which disk? "
$ SHOW DEVICE/FULL 'DISK_NAME'
```

   Use SYSMAN to execute this procedure on all nodes and view the result. What happens?

Solutions

1.

a. `$ RUN SYS$SYSTEM:SYSMAN SYSMAN> SET ENVIRONMENT /CLUSTER SYSMAN> DO SHOW SYSTEM`

b. `SYSMAN> DO SHOW USERS`

c. `SYSMAN> DO SHOW CLUSTER`

d. `SYSMAN> PARAMETERS SHOW /CLUSTER SYSMAN> PARAMETERS SHOW /SCS`

e. `SYSMAN> DO MCR NCP LIST EXECUTOR CHARACTERISTICS`

f. `SYSMAN> DISKQUOTA SHOW *`

2. Here is a command procedure that shows the time on all nodes:

```
$ RUN SYS$SYSTEM:SYSMAN
SET ENVIRONMENT /CLUSTER
DO SHOW TIME
```

3. Currently, SYSMAN cannot be used to run a command or utility that prompts for information. The INQUIRE command returns a null string, so the SHOW DEVICE command shows information about all devices on the system.

Laboratory Exercise 2

In this lab, you will practice writing restartable batch jobs. The batch job restart capability is especially useful in a cluster because if a node crashes, its batch jobs can restart immediately on other nodes. Write a command procedure that:

- Loops every minute
- Writes the name of the node it is running on to the batch log file
- Keeps a count of the number of times it has been through the loop, and
  - — Writes the number to the log file each time it is incremented
  - — Maintains the count even if the job is restarted
- Exits after 60 iterations

To maintain the count across restarts, your procedure must:

- Use the SET RESTART_VALUE command to save the number at every iteration.
- Examine the global symbol $RESTART at the beginning of the procedure. $RESTART has the value TRUE if the job has been restarted.
- If the job is restarted, extract the restart value from the global symbol BATCH$RESTART. This symbol contains the last value set by SET RESTART_VALUE before the job was aborted.

NOTE

This information, and help from your instructor, should allow you to complete the lab. The *Guide to Using VMS Command Procedures* also explains how to write a restartable batch job.

To be able to test this command procedure, your cluster must have a generic queue with at least two execution queues on different nodes assigned to it. To test this command procedure:

1. Submit your procedure to a local queue with SUBMIT/RESTART.
2. After it has looped a few times, use STOP/REQUEUE to stop the job. Observe that the job restarts.
3. Examine the log file to make sure the count was maintained across restarts.
4. Finally, submit your job to the generic queue.
5. If your instructor allows you to, shut down the VAX system on which the job is running.
6. If not, use STOP/RESET to stop the queue in which the job is running. The job should restart on another system.
7. Again, examine the log file.

Solutions

A possible solution to this exercise is the procedure V54_CLUMGT:COUNTER.COM, which follows.

```
$! COUNTER.COM -- restartable batch job
$!
$! If restarting, set up COUNT again
$!
$ SET NOVERIFY
$ IF $RESTART THEN GOTO AGAIN
$!
$! First time only: initialize the counter
$!
$ COUNT = 1
$LOOP:
$!
$! Hold the counter over a restart
$!
$ SET RESTART_VALUE='COUNT'
$!
$! Only action in procedure is to note system it is on
$!
$ NODE = F$GETSYI("NODENAME")
$ WRITE SYS$OUTPUT "The system we are running on is ", -
  "''NODE'"
$!
$! Show how often we've been round the loop
$!
$ WRITE SYS$OUTPUT "Been round the loop ", 'COUNT', " times"
$!
$! Wait, increment the counter and loop (don't loop forever)
$!
$ WAIT 00:01
$ COUNT = COUNT + 1
$ IF COUNT .EQ. 60 THEN EXIT
$ GOTO LOOP
$!
$! Only on restart: reset COUNT to value held over restart
$!
$ AGAIN:
$ COUNT = BATCH$RESTART
$ WRITE SYS$OUTPUT "This is a restart at loop ", "''COUNT'"
$ GOTO LOOP
$
```

Laboratory Exercise 3

This exercise can be annoying to other students. Your instructor may designate a certain time interval during which you may do this exercise; please refrain from using the REPLY command at other times.

1. Issue a REPLY command that sends a message to all users on the node to which you are logged in.

2. Issue a REPLY command that sends a message to all users except the ones on the node to which you are logged in.

3. Issue a REPLY command to a specific terminal on another node.

4. Log in to more than one node simultaneously. (If your terminal is on a terminal server that allows multiple sessions, create two sessions on different nodes. Otherwise, use SET HOST to log in to a different node, or use two terminals for this exercise.) Issue a REPLY command to send a message to your own user name, and verify that you receive the message on both nodes.

Solutions

In the following examples, substitute the actual names of nodes and devices in the VAXcluster system you are using.

1. `$ REPLY /USERS /NODE "This is message 1"`

2. `$ REPLY /USERS /NODE=(node1,node2,...) "This is message 2"`

3. `$ REPLY /TERMINAL=node$device "This is message 3"`

4. `$ REPLY /USER=user-name`

LOCATING VAXcluster PROBLEMS

Laboratory Exercise 1

In this exercise, you will use the MONITOR command to examine and compare CPU usage and disk I/O for two nodes of a cluster. You will use MONITOR recording files in playback mode to simulate live monitoring of data. You also summarize data from multiple recording files. Data was collected for this lab by submitting two command procedures to run as concurrent batch jobs on two VAX systems in a cluster. The command procedure that ran on node COORS was as follows:

```
$ MONITOR /BEGINNING=14:30 /ENDING=14:40 /INTERVAL=20 -
          /RECORD=CMON.DAT ALL_CLASSES
$ EXIT
```

The command file that ran on node LITE was as follows:

```
$ MONITOR /BEGINNING=14:30 /ENDING=14:40 /INTERVAL=20 -
          /RECORD=LMON.DAT ALL_CLASSES
$ EXIT
```

Each of these batch jobs collected data for all classes of statistics once every 20 seconds for the same 10 minute period. The batch job on COORS deposited its data in a file called CMON.DAT, and the batch job on LITE deposited its data in a file called LMON.DAT.

1. First, examine CPU activity. Replay five minutes of data in the MODES class for node COORS by typing the following command:

```
$ MONITOR /INPUT=V54_CLUMGT:CMON.DAT -
  /BEGIN=7-FEB-1985:14:30 -
  /END=7-FEB-1985:14:35 MODES /PERCENT
```

   What percentage of COORS' CPU time was idle during this period?

2. By summarizing data for both nodes COORS and LITE for the same period of time, it is possible to see how well the CPU load was distributed between the CPUs in this cluster during this five-minute period of time. Issue the following commands:

```
$ SET TERMINAL /WIDTH=132
$ MONITOR /INPUT=(V54_CLUMGT:CMON.DAT,V54_CLUMGT:LMON.DAT) -
  /BEGIN=7-FEB-1985:14:30 -
  /END=7-FEB-1985:14:35 -
  /SUMMARY=TT: MODES /PERCENT
```

   Was the CPU load distributed evenly between COORS and LITE over this period? On the whole, was the cluster overloaded?

3. The MONITOR utility also accepts wildcard characters (* and %) in file names. Try the following command:

```
% MONITOR /INPUT=V54_CLUMGT:%MON.DAT -
  /BEGIN=7-FEB-1985:14:30 -
  /END=7-FEB-1985:14:35 -
  /SUMMARY=TT: MODES /PERCENT
```

4. Next, investigate I/O activity. Measure the I/O activity on system COORS for the disk drives in the cluster that are accessible to COORS. To measure the rate of I/O operations per second for the disks available to COORS, type the following commands:

```
$ SET TERMINAL /WIDTH=80
$ MONITOR /INPUT=V54_CLUMGT:CMON.DAT -
  /BEGIN=7-FEB-1985:14:30 -
  /END=7-FEB-1985:14:35 -
  DISK /ITEM=OPERATION_RATE
```

   Which disk volumes sustained the most I/O over this period?

5. Another way of measuring the I/O load is to measure the lengths of I/O request queues. To examine the queue lengths for the same disk drives during the same period of time, use this command:

```
$ MONITOR /INPUT=V54_CLUMGT:CMON.DAT -
  /BEGIN=7-FEB-1985:14:30 -
  /END=7-FEB-1985:14:35 -
  DISK /ITEM=QUEUE_LENGTH
```

Do the queue lengths seem consistent with the I/O rates you saw when you entered the previous command?

NOTE

Remember that this information is from COORS' point of view. It does not reflect I/O request queues maintained by LITE, and thus is not cluster-wide information.

6. Now, examine the I/O operation rates over the entire cluster by typing these commands:

```
$ SET TERMINAL /WIDTH=132
$ MONITOR /INPUT=V54_CLUMGT:%MON.DAT -
/BEGIN=7-FEB-1985:14:30 -
/END=7-FEB-1985:14:35 /SUMMARY=TT: -
DISK /ITEM=OPERATION_RATE
```

7. Examine the queue lengths over the cluster with the following command:

```
$ MONITOR /INPUT=V54_CLUMGT:%MON.DAT -
  /BEGIN=7-FEB-1985:14:30 -
  /END=7-FEB-1985:14:35 /SUMMARY=TT: -
  DISK /ITEM=QUEUE_LENGTH
```

8. Is there a need to do load balancing among the disk volumes? If so, on which disk(s) should you relieve some of the I/O load, and to which disk(s) should you transfer some of the load?

Solutions

See your instructor if you need the solutions to these exercises.

Laboratory Exercise 2

1. At your terminal, monitor the CLUSTER class. Are there any systems with no idle CPU time? Are there any disks that are sustaining a high I/O rate?

2. Submit batch jobs that will simultaneously monitor all nodes in the VAXcluster system, or just those nodes your instructor tells you to monitor. Monitor at least the MODES and DISK /ITEM=ALL classes, for at least ten minutes.

   As a model for your command procedure, use either the commands at the beginning of the previous exercise or the examples in Module 6, Managing VAXcluster Operations.

3. If you see a problem that requires load balancing, how might you alleviate the problem? Fix the performance problem and monitor the systems again.

Solutions

See your instructor if you need the solution to this exercise.

Laboratory Exercise 3

Use the SHOW CLUSTER utility, HSC SETSHO, NCP, and other utilities to complete the exercises that follow. As you complete the exercises, fill in a chart with the names and numbers associated with your lab VAXcluster system. Your chart should contain information like that found in the Appendix of Module 4 of your Student Workbook.

1. Determine the following information about the VAXcluster nodes:
   a. What type of hardware does each node use?
   b. What type of port (CI, Ethernet, or both) does this node use?
   c. What is the CI port number of each node connected to the CI bus?
   d. What is the Ethernet hardware address of each satellite node?
   e. What is the DECnet node name of each active node?
   f. What is the DECnet address of each active node?
   g. What is the SCS node name of each node?
   h. What is the SCS system ID of each node?
   i. Which system disk does each node boot from? Which system root is on that disk?
2. Determine the following information about the VAXcluster formation:
   a. When was this cluster formed?
   b. When was the most recent state transition?
3. Determine the following information related to quorum and votes:
   a. How many votes are expected in this cluster?
   b. How many votes are required to have a quorum in this cluster?
   c. If there is a quorum disk, what is the quorum disk's name and how many votes does it contribute to the cluster?
   d. How many votes does each node contribute to the cluster?

4. Obtain the following information regarding all virtual circuits between the node you are currently logged into and other nodes in the cluster:

   a. CI port number of the remote node, if the circuit uses the CI bus

   b. Type of remote port associated with the circuit

   c. The number of connections currently supported by the circuit

   d. The state of the circuit

   e. The cable status of the circuit paths, if the circuit uses the CI bus

5. Determine, for each connection on each virtual circuit you identified above, the following information:

   a. The name of the local SCS process associated with the connection

   b. The name of the remote SCS process associated with the connection

   c. The state of the connection

6. For each node, determine the following information related to VAXcluster resources:

   a. The cluster-available mass storage devices connected to each node

   b. The mass storage devices on each node that are not cluster-available

   c. Whether there are any shadow sets, and which disks belong to each shadow set

   d. The generic queues enabled and which execution queues they serve

   e. The execution queues enabled

7. Determine the following information related to the DECnet network:

   a. What interconnect is used for DECnet traffic, and what is its circuit name?

   b. Which nodes are routers?

   c. Is a cluster alias defined? Which nodes use it?

Solutions

1. Use the following commands:

   a. `$ SHOW CLUSTER /CONTINUOUS`
      `Command> ADD HW_TYPE`

   b. `$ SHOW CLUSTER /CONTINUOUS`
      `Command> ADD RP_TYPE`

   c. `$ SHOW CLUSTER /CONTINUOUS`
      `Command> ADD RPORT`

   d. From boot node: `$ MCR NCP LIST NODE node-name CHARACTERISTICS`

   e. On each node: `$ SHOW NETWORK`

   f. `$ MCR NCP LIST NODE node-name`

   g. `$ SHOW CLUSTER /CONTINUOUS`
      `Command> ADD NODE`

   h. `$ SHOW CLUSTER /CONTINUOUS`
      `Command> ADD SYS_ID`

   i. For a satellite node, execute:

      `$ MCR NCP LIST NODE node-name CHARACTERISTICS`

      On a boot node; look at the load assist parameter.

      For a CI node, log in and display the values of logical names SYS$SYSDEVICE and SYS$SPECIFIC.

2. Use the following commands:

   a. `$ SHOW CLUSTER /CONTINUOUS`
      `Command> ADD FORMED`

   b. `$ SHOW CLUSTER /CONTINUOUS`
      `Command> ADD LAST_TRANSITION`

3. Use the following commands:

   a.
```
$ SHOW CLUSTER /CONTINUOUS
Command> ADD CL_EXPECTED_VOTES
```

   b.
```
$ SHOW CLUSTER /CONTINUOUS
Command> ADD CL_QUORUM
```

   c.
```
$ SHOW CLUSTER /CONTINUOUS
Command> ADD QD_NAME, CL_QDVOTES
```

   d.
```
$ SHOW CLUSTER /CONTINUOUS
Command> ADD VOTES
```

4.
```
$ SHOW CLUSTER /CONTINUOUS
Command> ADD CIRCUITS/ALL
```

5.
```
$ SHOW CLUSTER /CONTINUOUS
Command> ADD CONNECTIONS/ALL
```

6. Use the following commands:

   a.
```
$ SHOW DEVICE D
$ SHOW DEVICE MU
```

   b.
```
$ SHOW DEVICE /FULL D
$ SHOW DEVICE MU
$ SHOW DEVICE MT
$ SHOW DEVICE MF
```

   c.
```
$ SHOW DEVICE DS
```

   d.
```
$ SHOW QUEUE
```

   e.
```
$ SHOW QUEUE
```

7. Use the following commands:

   a.
```
$ MCR NCP LIST KNOWN CIRCUITS CHARACTERISTICS
```

   b.
```
$ MCR NCP LIST EXECUTOR CHARACTERISTICS
```

   c.
```
$ MCR NCP LIST EXECUTOR CHARACTERISTICS
```

Laboratory Exercise 4

This lab has two parts:

1. An introduction in which you practice using VAXsim software on data provided as part of the VAXsim installation kit.

2. A problem tracing session where you trace a problem on a two-node VAXcluster system. The data for this session was obtained by executing a command file, VAXSIMLOA.COM, on each processor in the cluster whose CPUs are known as COORS and LITE.

VAXSIMLOA.COM, which is also provided as part of the VAXsim installation kit, merely initializes the VAXsim database with information from an error log file. In this case, the SYS$ERRORLOG:ERRLOG.SYS files from both COORS and LITE were used to initialize a database for this lab.

Part 1: Introduction and Tutorial

For part 1 of this exercise, your instructor has provided you with the manual *Getting Started with VAXsimPLUS*. Read the introduction and then perform the steps for the General Session in chapter 2.

Part 2: Application in an Actual VAXcluster Environment

In this part of the exercise, you will use what you learned in part 1 to examine data from an actual VAXcluster configuration.

1. To inform VAXsimPLUS software about the names of the nodes in the cluster and identify for VAXsimPLUS the database for this cluster, type the following commands:

```
$ DEFINE VAXSIM$CLUSTER_COORS V54_CLUMGT:VAXSIM_COORS.DAT
$ DEFINE VAXSIM$CLUSTER_LITE V54_CLUMGT:VAXSIM_LITE.DAT
```

2. Now start VAXsimPLUS software by typing

```
$ VAXSIM
```

   and wait until VAXsimPLUS displays information about the system you are on and gives you the VAXsim> prompt.

3. Since you are going to examine data about a cluster which probably does not include the system you are on, it is necessary to remove information about this system from the VAXsimPLUS display database. To do so, type the command:

```
VAXsim> REMOVE name-of-system-you-are-on
```

   VAXsimPLUS should then tell you that there is no further information on record at this time.

4. Now insert into the VAXsimPLUS display database information about the cluster you are going to deal with by typing the command:

```
VAXsim> ADD COORS,LITE
```

VAXsimPLUS will then display a system level block diagram showing you the cluster consisting of COORS and LITE.

NOTE

As you progress further with this part of the lab, you should remember that COORS and LITE share devices. Consequently, whether you pursue information down the COORS tree or the LITE tree, be prepared to occasionally see the same devices and device names in both trees.

5. Track a disk drive problem on COORS. Start with the command:

```
VAXsim> SINCE 21-DEC-84:11:12AM\BEFORE 21-DEC-84:11:18AM
```

and identify which drive is having the problem, the type of drive it is, and try to describe what the failure is.

6. Now track a problem on a shared device to see how both VAX nodes in the cluster reported failures with this device.

   - Begin by typing the commands:

     ```
     VAXsim> TOP
     VAXsim> SINCE 25-JAN-85:8:00AM\BEFORE 29-JAN-85:5:00PM
     ```

     Notice that VAXsimPLUS points out potentially serious problems on both COORS and LITE.

   - Next, track down the COORS tree to see which device is having problems and note from the error detail level what those problems were and their frequency.

   - Now type the TOP command and then track down the LITE tree. Notice that it leads to the same device. Compare the problems and their frequency from the error detail level on LITE with what you observed when you tracked down the COORS tree.

7. Track a problem on LITE, starting with the commands:

```
VAXsim> TOP
VAXsim> SINCE 29-DEC-84:8:00AM\BEFORE 1-JAN-85:5:00PM
```

Even though the errors are reported as being soft (correctable) errors, note how many occurred in just the short period of time specified by the SINCE and BEFORE parameters you specified above.

Solutions

See your instructor if you need the solutions to this exercise.

Laboratory Exercise 5

In this exercise, you create a command procedure that checks the configuration of your cluster periodically to make sure all your hardware is working. Because the hardware redundancies in a cluster make it possible for equipment to fail without your knowledge, this procedure can be a useful tool. You can develop and test such a command procedure in several steps. Do as many of these steps as you have time for. There may be more than one way to do some of these steps; the hints give the methods used by the sample solution on the next page.

- Write a command procedure that automatically executes in batch at a certain time interval.
- Add statements to check for the existence of each node in the VAXcluster system each time the procedure executes.
  - — Hint: Use the F$GETSYI lexical function to check whether a certain VAX node is a member of the cluster.
  - — If you do not have DCL manuals available during the course, use the HELP command to get information on lexical functions.
- Add statements to send you mail if any node is not in the cluster.
- Add statements to check for the existence of each HSC controller in the VAXcluster system and to send you mail if any of them is unavailable.
  - — Hint: Direct SHOW CLUSTER output to a file. Look for the name of the HSC in the file and make sure the virtual circuit to the HSC is OPEN.
- Add statements that make sure each disk in the cluster is available and to send you mail if not.
  - — Hint: Use the F$GETDVI lexical function to find out whether a disk is on-line. It can also tell you whether its primary host is available and whether the disk is dual-pathed (and if so, whether the secondary host is available).

Solutions

There is a command procedure in V54_CLUMGT:CLUSTER_CHECK.COM that performs the given tasks for a sample cluster. Here is a listing of the procedure:

```
$! CLUSTER_CHECK -- VAXcluster Configuration Checker
$!
$! This command procedure verifies the hardware/software configuration of
$! a VAXcluster system. It runs periodically, and sends mail to SYSTEM if it notices
$! a problem.
$!
$! This procedure is set up to run on node VAXA. It is hard-coded to check
$! for nodes VAXB and HSC003, and disk device $255$DUA0:.
$!
$! This example demonstrates how to check for the existence of nodes and
$! devices. To make the command procedure more flexible, don't hard-code node
$! and device names, but maintain them in a file which this procedure
$! can read.
$!
$! First, submit self to execute every hour.
$! Be sure to specify the local queue explicitly.
$
$ SUBMIT /NOPRINT /NOLOG /NOTIFY /AFTER="+1:00" /QUEUE=VAXA_BATCH -
        'F$ENVIRONMENT("PROCEDURE")'
$
$! Check for presence of VAXB using F$GETSYI.
$
$ IF F$GETSYI ("CLUSTER_MEMBER", "VAXB") THEN GOTO MEMBER_DONE
$
$! If it's not present, send mail to SYSTEM.
$
$ MAIL/SUBJ="VAXB is not present" NL: SYSTEM
$MEMBER_DONE:
$
$! Check for presence of HSC003, with open virtual circuit:
$! First create temporary SHOW_CLUSTER$INIT to check circuit status, and
$! run SHOW CLUSTER.
$
$ OPEN/WRITE TEMPFILE SYS$SCRATCH:SHCL_INPUT.TMP
$ WRITE TEMPFILE "ADD CIR_STAT"
$ CLOSE TEMPFILE
$ DEFINE /USER SHOW_CLUSTER$INIT SYS$SCRATCH:SHCL_INPUT.TMP
$ SHOW CLUSTER /OUTPUT=SYS$SCRATCH:SHCL_OUTPUT.TMP
$ DELETE SYS$SCRATCH:SHCL_INPUT.TMP;
$
$! Read SHOW CLUSTER output, looking for HSC003 and OPEN on same line.
$
$ OPEN/READ TEMPFILE SYS$SCRATCH:SHCL_OUTPUT.TMP
$HSC_LOOP:
$ READ /END=HSC_NOT_FOUND TEMPFILE LINE
$ IF ((F$LOCATE ("HSC003", LINE) .LT. F$LENGTH (LINE)) .AND.          -
        (F$LOCATE ("OPEN", LINE) .LT. F$LENGTH (LINE))) THEN GOTO HSC_FOUND
$ GOTO HSC_LOOP
$
$! If no lines met this condition, send mail to SYSTEM.
$
$HSC_NOT_FOUND:
$ MAIL/SUBJ="HSC003 is not present" NL: SYSTEM
$HSC_FOUND:
$ CLOSE TEMPFILE
$ DELETE SYS$SCRATCH:SHCL_OUTPUT.TMP;
$
$! Verify that the device $255$DUA0: exists by using FGETDVI.
$
```

```
$ IF .NOT. FGETDVI ("$255$DUA0:","EXISTS") THEN GOTO DEVICE_NOT_FOUND
$
$! If it exists, verify that its host(s) also exist.
$
$ IF .NOT. FGETDVI ("$255$DUA0:","HOST_AVAIL") THEN GOTO HOST_UNAVAILABLE
$ IF FGETDVI ("$255$DUA0:","HOST_COUNT") .LE. 1 THEN GOTO DEVICE_DONE
$ IF FGETDVI ("$255$DUA0","ALT_HOST_AVAIL") THEN GOTO DEVICE_DONE
$
$! If primary host is available but secondary isn't, send mail to SYSTEM.
$
$ MAIL/SUBJ="$255$DUA0: secondary host unavailable" NL: SYSTEM
$
$! If primary host is unavailable, send mail to SYSTEM
$
$HOST_UNAVAILABLE:
$ MAIL/SUBJ="$255$DUA0: host unavailable" NL: SYSTEM
$ GOTO DEVICE_DONE
$
$! If device doesn't exist, send mail to SYSTEM
$
$DEVICE_NOT_FOUND:
$ MAIL/SUBJ="$255$DUA0: does not exist" NL: SYSTEM
$ GOTO DEVICE_DONE
$DEVICE_DONE:
$
$ EXIT
```

Laboratory Exercise 6

In this lab you cause and observe VAXcluster problems. Many of the problems you cause are disruptive to others working on the cluster. Do not start this lab without permission from your instructor. Make sure at least several terminals are enabled as operator terminals and be sure to read all messages sent to the console terminals (including the HSC controller).

1. For a satellite node:

   a. On the boot node, run CLUSTER_CONFIG.COM to remove the satellite node from the cluster.

   b. Attempt to reboot the satellite into the cluster and observe the results on the boot node and the satellite.

   c. On the boot node, run CLUSTER_CONFIG.COM to add the satellite node back into the cluster. Reboot the satellite so that it joins the cluster.

2. For a node with MSCP served disks:

   a. From another node, create an editing session of a file on a disk MSCP served to the cluster by a satellite node you are about to shut down. Do not terminate the editing session.

   b. Shut down the node using SYS$SYSTEM:SHUTDOWN.COM. Do not invoke the site-specific shutdown procedure.

   c. On the remote system, type SHOW DEVICE D.

   d. After shutdown is complete, bring the node back into the cluster.

   e. Recover from the problem.

3. For the CI cables:

   a. Disconnect and leave disconnected a single CI cable from a CPU.

   b. Disconnect for 20 seconds and then reconnect the second CI cable to the CPU.

   c. Disconnect for 2 minutes and then reconnect the second CI cable to the CPU.

   d. Connect back the first CI cable for the CPU.

   e. Cross the CI cables for a single CPU.

   f. Cross the CI cables for the second CPU.

   g. Uncross the CI cables for the first CPU.

h. Uncross the CI cables for the second CPU.

i. Disconnect a single CI cable to the HSC unit.

j. Disconnect the second CI cable to the HSC unit.

k. Connect back the cables to the HSC unit.

4. For the CI780/CI750:

   a. Power down the CI780/CI750 for an active node.

   b. Observe the console messages.

   c. Power up the CI780/CI750.

5. For an HSC unit:

   a. Place the SECURE/ENABLE switch in the ENABLE position.

   b. Hold in the FAULT switch and press the INIT switch.

   c. Recover.

6. For an active node:

   a. Modify the SYSGEN parameter VOTES to 5.

   b. Remove the CPU from the cluster.

   c. Boot the CPU with votes equal to 5.

   d. Use SHOW CLUSTER to observe the CLUSTER display.

   e. Remove the CPU from the cluster.

   f. Observe the results.

   g. Reboot the CPU and return its VOTES value to an appropriate number.

7. For an active node (provided that there is another CPU of the same type):

   a. Replace the console media of one CPU with the console media of another cluster node of the same type.

   b. Reboot the CPU containing the wrong console media.

   c. Replace the correct console media.

   d. Reboot the CPU.

8. For a cluster with a quorum disk:

   a. Take the quorum disk off-line.

   b. Observe the results.

   c. Put the quorum disk back on-line.

Solutions

No solutions necessary.

digital™

Printed in The Netherlands